Coordination in hierarchical pickup and delivery problems using delegate multi-agent systems

Rutger Claes
DistriNet Labs
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3000 Leuven, Belgium
Email: rutger.claes@cs.kuleuven.be

Tom Holvoet
DistriNet Labs
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3000 Leuven, Belgium
Email: tom.holvoet@cs.kuleuven.be

Jelle Van Gompel
DistriNet Labs
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3000 Leuven, Belgium
Email: jelle.vangompel@student.kuleuven.be

Abstract—**Pickup and delivery problems are a generalization of the planning problem faced by transport companies. Large logistics providers often employ a hierarchical 'hub and spoke' overlay network to connect pickup and drop off sites. Hierarchical pickup and delivery problems pose two major challenges: (1) determining suitable routes from the origin to the destination of packages through the logistics providers' network, and (2) allocating the resources that are required for pickup, transporting and delivering the packages along such a route.**

By combining traditional resource allocation techniques with swarm algorithms, the approach in this paper offers a decentralized solution to pickup and delivery problems in hierarchical environments. Resources are scheduled locally at the node from which they operate, resulting in a distribution of many local resource schedules. A swarm approach called delegate multi-agent systems is used to extract information from relevant localized schedules and combine them in consistent global paths. The ant-like agents in these delegate multi-agent systems also redistribute feedback from the path planning mechanism to the decentralized resource scheduling mechanism.

Results obtained by this hybrid approach show that it outperforms greedy and static alternatives.

I. INTRODUCTION

This paper proposes a decentralized mechanism to control and coordinate transportation resources, such as trucks and airplanes, in hierarchical pickup and delivery problems. The general pickup and delivery problem is described in [1]. The approach we present here uses a biologically inspired swarm algorithm, called *delegate multi-agent systems*, to coordinate a distributed heuristic scheduling algorithm.

Large logistics providers organize their transportation network in a hierarchical 'hub and spoke' overlay network[2]. In order to reach their destination, packages are routed from terminal to terminal in this overlay network. If the destination of a package falls outside the region of a particular terminal, it is forwarded to a terminal on a higher level. This process is repeated until the package reaches a terminal covering its destination or the highest level of the hierarchy is reached. At the highest level, the top-level hubs are connected in a mesh-like fashion. An example of hierarchical hub and spoke overlay network is depicted in Figure 1.

Packages need to be routed through this hierarchical network in order to reach their destination. Between each level

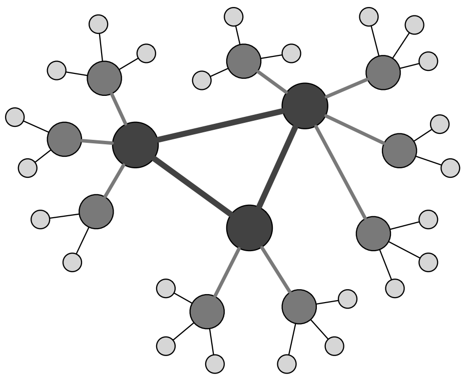

Fig. 1. A hierarchical hub and spoke overlay network consisting of three levels. The nodes at the top level are connected in a mesh-like fashion.

in the hierarchy, transportation resources need to be allocated in order to actually transport the packages up and down the hierarchy. Scheduling these resources is a difficult process due to a number of challenges, including the dynamic nature of the transportation demand, the scale of the hierarchical overlay network, and unforeseen events - e.g. the recent eruption of the Eyjafjallajökull that can disrupt global transportation, severely influencing the schedule.

The scale and dynamics of the problem make it hard for centralized one-shot planning algorithms to provide all resources with an efficient and up to date schedule. A decentralized solution that takes into account the problem dynamics and focusses on a continuously updated schedule, while probably not globally optimal, seems a better fit for this problem.

The approach presented in this paper combines an offline scheduling algorithm with a biologically inspired multi-agent coordination and control mechanism. The approach exploits the hierarchical nature of the overlay network. Every node in the hierarchical network is responsible for a number of transportation resources. These resources include delivery trucks and - at higher levels in the hierarchy - airplanes. Each node of the network is responsible for the schedule of its resources, improving the scalability of the approach by eliminating single points of failure and bottlenecks.

The constraints used in the scheduling algorithms are distributed across the nodes in the network using a multi-agent system. By propagating this constraint information through the network, the different scheduling processes are synchronized. This leads to a more efficient usage of transportation resources.

Outline of the paper: In the next section we outline a hybrid coordination and control mechanism for hierarchical pickup and delivery problems. We provide details on how the swarm-based algorithm and heuristic scheduling algorithm work together to enable a dynamic coordination mechanism. Next we relate our approach with existing work. After that we compare our solution with two base-line providing alternatives. We end this paper with some concluding remarks.

II. Coordination and Control in Hierarchical Pickup and Delivery Problems

The approach we present in this paper combines a traditional heuristic planner with a coordination and control mechanism using delegate multi-agent systems [3]. The approach is package-centric, in that it tries to find suitable routes for packages in a bottom up approach and builds the resources schedules based on package demands.

A. Agent based modeling

The approach that we study in this paper models the problem domain as a set of agents, cooperating together in a multi-agent system. Packages are represented by *package agents*, depots are represented by *depot agents*. Together these agents decide upon the route that a package will traverse through the logistics providers network, they decide upon the allocation of transportation resources such as planes and trucks and they determine the schedule of these resources.

As mentioned, the solution presented here takes a bottom up approach. Finding a suitable route through the hierarchical overlay network is the responsibility of the package agents. When entering the system, every package is assigned to a package agent. Every package agent is responsible for one package and travels virtually alongside the package until it reaches its destination. Finding a route through the overlay network and the resources necessary to transport the package is a task the package agent accomplishes using *delegate multi-agent systems*.

Delegate multi-agent systems are multi-agent systems composed out of lightweight agents. Together these lightweight agents can perform tasks on behalf of other, often more complex, agents. The individual behavior of the lightweight agents is usually rather basic. However, because the agents cooperate with each other, behavior satisfying the needs of the controlling agent emerges from their interactions. Because the behavior attributed to these light-weight agents often resembles that of behavior exhibited by ants, they are referred to as ant-like agents or simply ants.

In the proposed solution, the package agents use delegate multi-agent systems to perform two tasks. First, a delegate multi-agent system is used to select the most suitable route

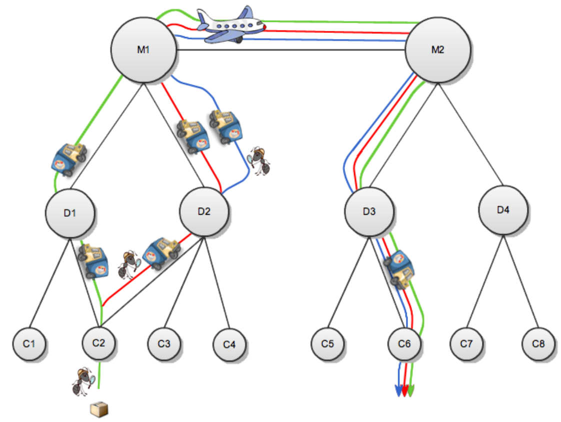

Fig. 2. Exploration ants explore the hierarchical network. On arrival the exploration ants have a schedule including arrival and departure times of transportation resources. In this figure, exploration ants search for feasible paths between node *C2* and *C6*.

from the packages current position to its destination. The ant-like agents forming this delegate multi-agent system do this by generating and examining several feasible routes through the hierarchical overlay network.

The second task performed by a different delegate multi-agent system on behalf of the package agent is distributing the constraints they have gathered along this selected route to all relevant scheduling processes. Both of these tasks are discussed more in depth in the following sections.

a) Selecting routes through the overlay network: Package agents are responsible for finding a suitable route through the overlay network. No global knowledge is assumed, every node in the hierarchical overlay network only knows about its immediate neighbors up and down the hierarchy. This assumption benefits the scalability of the approach. Instead of starting to search the overlay network in order to find a suitable path to its destination, the package agent delegates this task to a delegate multi-agent system.

Because of the role they play in the mechanism, the ant-like agents in this delegate multi-agent system are referred to as *exploration ants*. They traverse the overlay network in search for the destination of the package their package agent represents. Figure 2 depicts such an exploration.

The search tree roamed by the exploration ants is an extension of the hierarchical overlay network shown in Figure 2. Between two connected nodes in the hierarchical overlay network there can be any number of connections based on the current schedule of the transportation resources. Figure 2 shows two trucks between nodes *D2* and *M1*. As these trucks are likely to have different schedules, they form two distinct paths. The arrival time of these two trucks influence the available resources further in the path. The choice of transportation resource between *D2* and *M1* may well determine the flight the package will be on between *M1* and *M2*.

At every node in the network they come across, they query the schedules of the available transportation resources managed by the local *depot agent* responsible for the depot at that particular node. This information is used by the ants to put together a *longitudinal schedule* for the package. A longitudinal schedule describes the path a package must follow to get from its origin to its destination. It not only includes the nodes through which the package must traverse, but also the transportation resources that will transport the package between these nodes. For every transport, the longitudinal schedule includes the departure and arrival time of the transport resource. The longitudinal schedule is only valid when the arrival time at a particular node falls before the departure time at that same node.

All ants arriving at the packages destination carry with them such a complete and valid longitudinal schedule. Using a heuristic, an optimal schedule is chosen from the set of discovered schedules. This represents the *intention* of the package agent. The arrival and departure times contained in the chosen schedule that ensure its validity now become constraints. If the package arrives too late at a certain node, it jeopardizes the remainder of its schedule.

Using ants to explore the environment for feasible routes reduces the responsibilities of the package agent. Multiple ants are sent out simultaneously, thus exploring the environment in parallel.

b) Coordinating package transportation constraints: As outlined in the previous section, the schedules for transportation resources are built and maintained by the *depot agents*. These schedules can be seen as *cross-sectional* schedules: schedules that describe the transports of a number of transportation resources servicing a number of connections. Visiting exploration ants can use the information contained in these schedules to construct their longitudinal schedules, thus chaining transports from the different cross-sectional schedules on their path together.

The depot agents try to optimize their schedules taking into account current intentions of all package agents intending to pass through their node. This is done as follows. When the best path is chosen at the end of the exploration phase, a *intention ant* is sent back across this chosen path. This ant informs the depot agents on the path of the pending visit from the package. This feedback information helps the depot agent in two distinct ways: (1) It helps the depot agent take into account the capacity of its transportation resources. The depot agent could limit the options presented to future exploration ants if certain scheduled transportation resources are already at maximum capacity. And (2) the depot agent learns which parts of the current schedule are critical to the current package schedules.

Because the exploration ants chain together different schedules, they introduce dependencies. If the depot agents learn about these dependencies, they can take them into account in their ongoing efforts to optimize their local schedules. Looking back at Figure 2 we see that flights between *M1* and *M2* depend on the arrival times of packages at *M1*. The responsible depot agent can change the schedules between *M1*, *D1* and *D2*, but it has to guarantee the on time arrival of packages wanting a flight out of *M1*.

B. *Continuous adaptation of the schedule*

To summarize, the interaction between the swarm algorithm initiated on behalf of a package agent and the localized scheduling algorithm employed by the depot agents can be seen as encompassing the following steps:

1) Exploration ants roam the hierarchical overlay network in search of feasible paths.
2) Depot agents inform passing exploration ants about the schedules and limitations of their resources.
3) Exploration ants finish their exploration phase and gather at the packages destination.
4) One possible path is selected based on heuristic functions.
5) Information about the package schedule is carried back across the path by an intention ant.
6) Depot agents use the information they learn to inform future exploration ants and optimize their local schedule.

These steps are repeated until the package triggering the explorations has reached its destination. This interaction allows depot agents to continuously optimize their local schedule, while still taking into account global constraints. Depot agents can alter and extend their schedules as long as they manage to adhere to the constraints brought back to them by the intention ants. In order for the package agents to take advantage of possible improvements in the local schedules, the exploration phase is repeated regularly. Because of this ongoing exploration, package schedules can change over time. This mechanism of continuous planning also improves the robustness of the schedule to changes in the overlay network topology or unforeseen delays.

Exploration ants will notice changes in the overlay network topology while exploring the network and, while doing so, come up with alternative paths. If the exploration ants find a better schedule than the one they are currently committed to, because of optimizations done by the depot agent, the package can change its intention. This introduces a new problem: *stale constraints*. The constraints carried back by a intention ant are no longer relevant as the packages schedule has changed. To solve this problem of stale constraints, the approach borrows from nature and uses the principle of *pheromones*.

Social insects often use evaporating chemicals, called pheromones, to communicate with each other. The pheromones are deposited in the environment by one insect and can be detected by others. Information is passed through the intensity of the pheromone as well as by its location in the environment. Ants for example, use pheromones to mark the route to newly discovered food supplies. The evaporating nature of pheromones ensures freshness of information. Information that is not regularly reinforced simply evaporates. When applied to the constraints, old constraints evaporate unless intention ants reinforce them regularly. As the repeating of the exploration phase triggers intention ants to go back over the optimal path, the information is reinforced as long as the path remains optimal and starts to evaporate as soon as a better alternative is discovered.

The use of pheromones in optimization problems or transport coordination is a proven pattern used in many other applications [4], [5], [6], [7].

The principle of evaporating constraints allows for easy intention switching based on local optimizations while ensuring correct and updated constraints throughout the network.

III. Related work

In this section we relate our approach with existing work. Related work can be found in many domains, including traffic coordination and solutions to package delivery problems. We try to relate our approach with work from different domains.

a) Travel Time prediction for Dynamic Routing using Ant Based Control: The approach presented in this paper bares resemblance with that described in [7]. Tatomir et al. describe a vehicle routing approach in a hierarchical traffic network using Ant Based Control mechanism. This approach relates to our work in two ways: the exploitation of hierarchical structures in the problem domain to facilitate coordination and the use of biologically inspired ant-based coordination mechanisms.

In the paper, the road network is modeled as a hierarchical network. The nodes in this hierarchical network are the cities. These high level nodes are connected through motorways. Every city node, internally, consists of a separate traffic network. Different motorway exits serve as gateway between the global network and the city networks. The decision to threat the environment as a hierarchical network is not forced, it is a choice made to facilitate route finding in the large network that is the traffic network. In our approach, companies make the decision to structure their overlay network as a hierarchical network because of economic reasons, not algorithmic reasons. The benefits, however, when route finding is concerned remain largely the same.

Tatomir et al. also describe the use of ants in their work. The equivalent of an *exploration ant* is a *forward ant*, the *intention ants* can be seen as a variant of the *backward ant*. The classification of ants used by Tatomir et al. is based on the direction in which the travel. In our work, such as [3], the classification is based on the task the ant performs.

The use of ants in [7] and in our approach differs greatly. In the Ant Based Control mechanism, ants are mainly used to generate routing information in the environment. Forward ants are periodically sent out from every node in a sector to a random other node. The forward ant will search the network for its destination, traveling from node to node. When a forward ant reaches its destination it triggers a backward ant that will track back across the path of the forward ant, leaving clues in the environment about the path one can follow to reach the destination. In this approach, ants are used to generate and spread routing information in the network. Ants are not, contrary to our approach, used to actively search and select routes nor for informing the network of pending visits.

Later work by Tatomir, such as [8], follows the paradigm proposed in [3] more closely and has the concept of vehicle agents actively sending out the forward ants. This is more in line with the approach taken in this paper.

b) Route computation in large route networks: a hierarchical approach: In [9] the benefits of hierarchical networks with respect to route finding are also described. The authors, Jagadeesh and Srikanthan, focus on route finding in large road networks. While not using ants, the approach bares some resemblance to the approach described in this paper. The search for a route to the destination is split up in a number of smaller searches. Each of this smaller search action only looks at one level of the hierarchy. This somewhat resembles the *exploration ants* in their search for suitable transportation resources at every level of the hierarchy they traverse. The goal of both approaches is identical: avoiding searching through the global network because of its size.

IV. Evaluation of the proposed approach

In this section we offer an evaluation of our approach. First we outline the setup in which we conducted our experiments. Next we detail the alternatives we compare our approach with. And finally, we present the results of our experiments with our proposed approach and the alternatives.

A. Experiment setup

Fig. 3. The three countries in the experimental setup.

In our experiments, we assume a hierarchical overlay network resembling that of Figure 1 spanning three different countries, namely Belgium, The Netherlands and England. The main airports of these three countries represent the hubs at the highest level in the overlay network. Each of the three countries has 30 depots located at the 30 largest cities of that country. These 30 depots are clustered in 5 clusters using a k-means clustering algorithm and the central depot of each of the 5 clusters is chosen as main regional depot.

The hierarchy thus consists of three levels, as shown in Figure 3. At the top the three airports are connected. These three airports are then connected to the 5 regional depots. The regional depots are each connected to other local depots in their respective clusters.

The connections between the different local depots, the regional depot and the top level hub consists of highways. The specific topology for each of the three countries is recreated using information from OpenStreetMaps. At the top level, the main hubs are connected to each other through flight routes.

When evaluating our approach the focus lies on two performance indicators. The first indicator is the transportation cost. The second indicator is the percentage of packages delivered on time. A penalty is assumed whenever a package is delivered too late. This penalty represents a loss of customers due to a bad reputation or a contractual penalty to be payed to the customer when deadlines are missed. In these experiments, a loss of 100 packages is assumed for every package that is delivered too late.

The profit of a package is assumed to be 50 units. The cost of a delivery truck is assumed to be .50 units per kilometer, that of an airplane is 10 times higher, 5 units per kilometer. Setting these profits and costs is essential to decide on the trade-off between an increase in costs and performance.

In order to evaluate our approach we compare it with two alternative coordination mechanisms. These alternatives only serve as a baseline in both performance and cost, and are not representative of the current state of the art in planning.

a) Greedy algorithm: The greedy approach represents the baseline in maximum performance using a naive approach. In the greedy algorithm transportation resources are assigned immediately to every new package that appears. Resource reuse is supported, but is not actively pursued, meaning that when a delivery truck drives towards an existing package it also picks up packages that happen to appear at the same location before the truck arrives there. Such a new package can piggyback on the already scheduled resource and will not call for a new resource.

From this description, it is clear that the greedy approach results in a very high delivery rate as packages are picked up as soon as possible. The approach, however, will suffer from a very high transportation cost due to the ad hoc scheduling of resources.

b) Static assignment: The static assignment represents a baseline in transportation costs. In this approach a best effort static schedule is drawn up, continuously deploying an average number of transportation resources. During the experiment the schedule remains fixed. Because of this lack of adaptation, the static assignment performs badly when looking at the package delivery rate as the static schedule is unable to cope with increasing loads. The static assignment, however, keeps the costs at a minimum while still trying to deliver as much packages as possible within the predefined timespan.

B. Experimental results

Graphs shown in this section show the results obtained using all three coordination mechanisms. All points shown are averages obtained by repeating the same scenario 15 times. Scenarios are determined by the transportation mechanism employed and the number of packages introduced in the system. The experiments are conducted using a simulation framework [10] simulating the transport of the packages on a road network level.

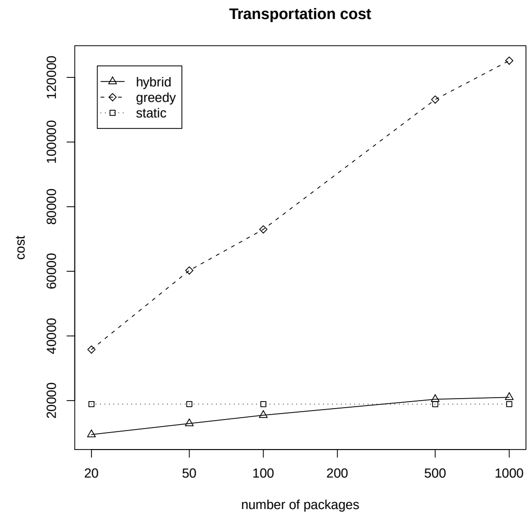

Fig. 4. The total transportation cost in function of the number of packages for all three strategies.

Figure 4 shows the transportation cost for all three coordination mechanisms. The transportation costs of the greedy algorithm are, as expected, much higher than the two other mechanisms. The transportation cost of the static assignment remains constant regardless of the increase in packages. The transportation costs generated by the hybrid approach increases as the number of packages needing transport increases, but remains comparable with those of the static assignment.

The graph shown in Figure 5 illustrates the difference in package delivery rates. A package is considered delivered when it arrives at its destination within the predefined timespan of 24 hours. Figure 5 shows that the greedy algorithm manages to deliver all packages before the deadline. As such, the greedy algorithm sets the baseline in terms of achievable performance. The hybrid approach presented in this paper continuously outperforms the static assignment. As the number of packages increases, the percentage of delivered packages stays around 99.8%.

The two previous results can be combined to examine the trade-off between cost and performance. Figure 6 shows the simulated profit for all three scheduling mechanisms. It shows that the static and hybrid approach are comparable in terms of profit and that both outperform the greedy algorithm. Figure 6 does not take into account the penalties imposed because of missed delivery deadlines. Figure 7 does include these penalties. In this graph it is shown that because of the missed deadlines, the estimated profits for the static assignment are lower than those obtained with the hybrid approach.

V. Conclusion

In this paper we have presented a novel approach to routing packages through a hierarchical delivery overlay network. The proposed approach uses a combination of biologically inspired swarm-algorithms, in the form of delegate multi-

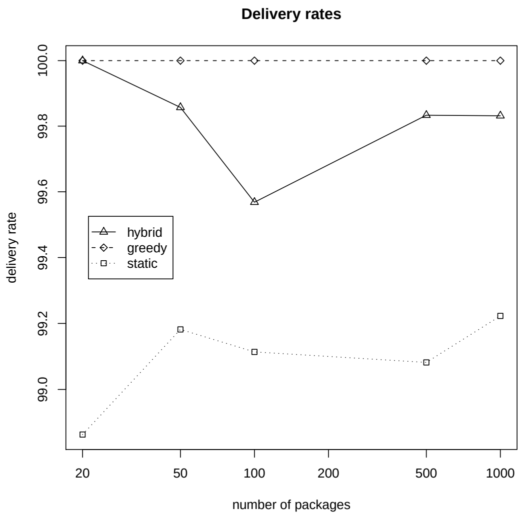

Fig. 5. The percentage of deliveries that reached their destination before the deadline.

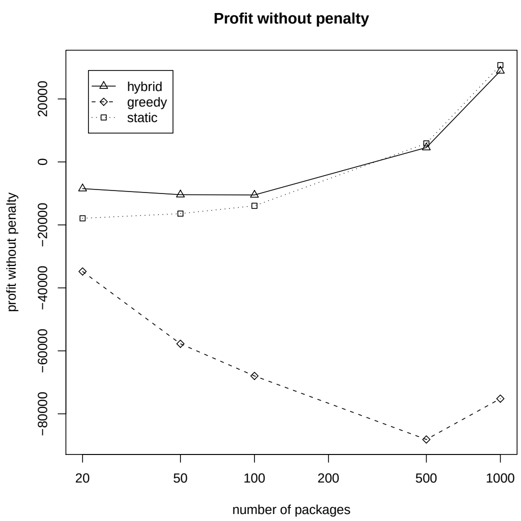

Fig. 6. The simulated profit in function of the number of packages for all three strategies.

agent systems, and traditional heuristic scheduling algorithms to achieve a decentralized adaptive coordination mechanism.

The evaluation of the coordination mechanism shows that it manages to successfully trade of transportation costs and performance. Because the approach presented here is distributed and takes advantage of the hierarchical nature of the distribution overlay network, the approach is expected to scale beyond the limit test case used in the evaluation.

The work presented in this paper is still at an early stage and more experiments must be conducted to further evaluate the proposed approach in even larger scenarios. The initial results, meanwhile, seem promising. The impact of different types of dynamics and scaling are still to be investigated. A benchmark

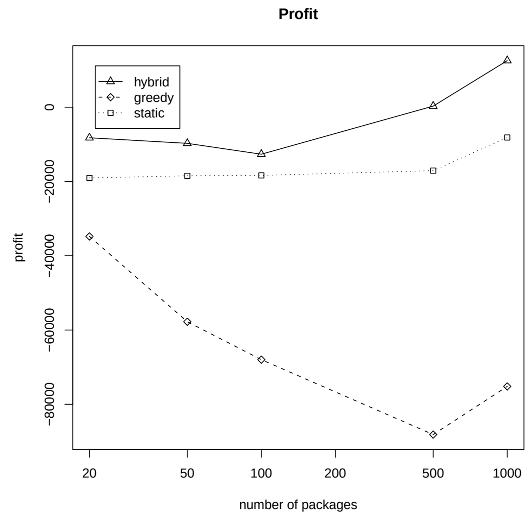

Fig. 7. The simulated profit with penalties for not delivering packages before their deadline taken into account.

of the proposed approach comparing it with other, state-of-the-art scheduling mechanisms under different scenarios is in order.

Acknowledgment

This research was funded by the IWT - SBO project MASE (project no. 060823) and by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund K.U.Leuven.

References

[1] M. W. P. Savelsbergh, "The general pickup and delivery problem," *Transportation Science*, vol. 29, no. 1, 1995.

[2] J. P. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*, 2nd ed. Routledge, 2009.

[3] T. Holvoet and P. Valckenaers, "Exploiting the environment for coordinating agent intentions," *Environments for Multi-Agent Systems III, LNAI 4389*, pp. 51–66, 2007.

[4] Y. Ando, O. Masutani, H. Sasaki, H. Iwasaki, Y. Fukazawa, and S. Honiden, "Pheromone model: Application to traffic congestion prediction," *Engineering Self-Organising Systems*, pp. 182–196, 2006.

[5] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[6] W. Narzt, G. Pomberger, U. Wilflingseder, O. Seimel, D. Kolb, J. Wieghardt, H. Hortner, and R. Haring, "Self-organization in traffic networks by digital pheromones," *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 490–495, 30 2007-Oct. 3 2007.

[7] B. Tatomir and L. Rothkrantz, "Hierarchical routing in traffic using swarm-intelligence," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 230–235, Sept. 2006.

[8] B. Tatomir, L. J. Rothkrantz, and A. C. Suson, "Travel time prediction for dynamic routing using ant based control," in *Proceedings of the 2009 Winter Simulation Conference*, M. Rosetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, Eds. IEEE Computer Society, 2009, pp. 1069–1078.

[9] G. R. Jagadeesh, "Route computation in large road networks: a hierarchical approach," *IET Intelligent Transport Systems*, vol. 2, no. 3, 2008.

[10] J. Van Gompel, B. Tuts, R. Claes, M. Cruz, and T. Torres, "MAS-DisCoSim 4 PDP: a Testbed for Multi-Agent Solutions to PDPs," in *AAMAS '10: Proceedings of the ninth international joint conference on Autonomous agents and multiagent systems*, 2010.